

# Optimasi Penjadwalan Perkuliahan dengan Menggunakan Algoritma Genetika (Studi Kasus : AMIK MDP, STMIK GI MDP dan STIE MDP)

Vinny Witary (vinny.wit25@gmail.com), Nur Rachmat (rachmat.nur91@gmail.com)  
Inayatullah (inayatullah@stmik-mdp.net)  
Jurusan Teknik Informatika  
STMIK GI MDP

**Abstrak :** Penjadwalan perkuliahan pada suatu perguruan tinggi atau universitas atau dikenal dengan sebutan *University Timetabling Problems* (UTP) merupakan hal yang rumit dan memakan waktu yang cukup lama karena harus mengkombinasikan antara mata kuliah yang ditawarkan, kesediaan dosen, dan ruangan. Saat ini, penjadwalan yang dilakukan pada AMIK MDP, STMIK GI MDP dan STIE MDP masih dilakukan secara konvensional sehingga penulis mencoba menyelesaikan masalah penjadwalan dengan membuat suatu aplikasi penjadwalan perkuliahan secara otomatis dengan menggunakan algoritma genetika sebagai metode optimasinya. Algoritma genetika adalah suatu algoritma pencarian yang berbasis pada mekanisme seleksi alam dan genetika. Algoritma genetika merupakan salah satu algoritma yang sangat tepat dalam menyelesaikan masalah optimasi kompleks, yang sulit dilakukan oleh metode konvensional. Aplikasi ini dibuat dengan bahasa pemrograman PHP, HTML, dan CSS pada *framework* CodeIgniter, serta menggunakan MySQL sebagai sistem *database*. Metodologi pengembangan sistem yang digunakan dalam pembuatan aplikasi ini yaitu *Rational Unified Process* (RUP). Hasil pengujian menunjukkan bahwa aplikasi ini mampu menghasilkan suatu penjadwalan perkuliahan yang optimal tanpa ada jadwal yang konflik.

**Kata kunci :** *University Timetabling Scheduling*, algoritma genetika, optimasi, PHP, HTML, CSS, *framework*, CodeIgniter, MySQL, *database*, *Rational Unified Process*.

**Abstract :** *The lectures scheduling at a college or university, known as University Timetabling Problems (UTP) is complicated and takes a long time because we have to combine the offered subjects, the available lecturers, and the rooms. Currently, the scheduling at AMIK MDP, STMIK GI MDP and STIE MDP was done conventionally so the author tried to solve this problems by making an automatically lectures scheduling application using genetic algorithm optimization method. Genetic algorithm is a search algorithm based on the mechanism of natural selection and genetics. Genetic algorithm is one of the very precise algorithm in solving a complex optimization problems, which so difficult for conventional methods. This application was made by PHP programming language, HTML, and CSS on CodeIgniter framework, and MySQL as the database system. System development methodology that used to build this application was the Rational Unified Process (RUP). The test results indicated that the application is capable for generating an optimal lectures schedule without conflict schedule.*

**Key Words :** *University Timetabling Scheduling*, genetic algorithm, optimization, PHP, HTML, CSS, *framework*, CodeIgniter, MySQL, *database*, *Rational Unified Process*.

## 1 PENDAHULUAN

Penjadwalan perkuliahan dalam suatu kampus merupakan kegiatan yang rumit dan memakan waktu yang cukup lama. Permasalahan ini sering disebut dengan *University Timetabling Problems* (UTP). Proses penjadwalan yang selama ini dilakukan masih secara manual sehingga

penulis berinisiatif untuk membuat suatu aplikasi yang dapat melakukan penyusunan penjadwalan perkuliahan yang optimal secara otomatis. Aplikasi ini dibuat agar dapat mencari suatu solusi agar tidak terjadi konflik antara jadwal yang satu dengan jadwal lainnya serta konflik pada penggunaan ruang kelas dikarenakan jumlah ruang kelas yang terbatas.

Salah satu metode yang dapat digunakan dalam pemecahan masalah ini yakni dengan menggunakan pendekatan algoritma genetika. Algoritma genetika merupakan algoritma yang berusaha menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan masalah (*problem solving*). Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi optimal di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik berikutnya yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau yang disebut dengan *fitness*.

## 2 LANDASAN TEORI

### 2.1 Definisi Teknik Optimasi

Optimasi adalah suatu pendekatan normatif untuk mengidentifikasi penyelesaian terbaik dalam pengambilan keputusan dari suatu permasalahan. Penyelesaian permasalahan dalam teknik optimasi diarahkan untuk mendapatkan titik maksimum atau titik minimum dari fungsi yang dioptimumkan. Tujuan dari optimasi adalah untuk meminimumkan usaha yang diperlukan atau biaya operasional dan memaksimalkan hasil yang diinginkan. Jika usaha yang diperlukan atau hasil yang diharapkan dapat dinyatakan sebagai fungsi dari peubah keputusan, maka optimasi dapat didefinisikan sebagai proses pencapaian kondisi maksimum dan minimum dari fungsi tersebut. Unsur penting dalam masalah optimasi adalah fungsi tujuan, yang sangat bergantung pada sejumlah peubah masukan. Peubah-peubah ini dapat tidak saling bergantung atau saling bergantung melalui satu atau lebih kendala.

### 2.2 Algoritma Genetika

Algoritma genetika pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Algoritma genetika adalah simulasi

dari proses evolusi Darwin dan operasi genetika atas kromosom.

Sebelum memanfaatkan algoritma genetika, hal yang harus dilakukan adalah menyandikan solusi dari masalah yang diberikan ke dalam kromosom pada algoritma genetika dan membandingkan nilai *fitness*-nya. Sebuah representasi algoritma genetika yang efektif dan nilai *fitness* yang bermakna adalah kunci keberhasilan dalam aplikasi genetika.

Pengkodean yang akan digunakan pada penjadwalan perkuliahan ini adalah pengkodean nilai. Pemilihan pengkodean nilai karena jika dilakukan pengkodean biner akan membuat barisan bit semakin kompleks, dimana suatu barisan bit yang panjang sehingga diperlukan pengkodean ulang untuk mendapatkan nilai sebenarnya dari aturan yang direpresentasikan.

Mahasiswa tidak dimasukkan dalam pengkodean kromosom, karena mahasiswa diberi kebebasan untuk memilih mata kuliah yang ditawarkan. Pemberian tugas mengajar suatu mata kuliah kepada dosen telah dilakukan sebelum pembuatan jadwal. Secara sederhana, jika tidak ada dosen yang mengajar suatu mata kuliah maka mata kuliah tersebut tidak ditawarkan.

Algoritma genetika bekerja dengan menggunakan pendekatan random, sehingga nilai-nilai yang dihasilkan adalah nilai random. Pada kasus penjadwalan dengan model genetika ditujukan untuk mendapatkan kombinasi yang tepat antara variabel dosen, waktu, dan ruang yang tidak saling konflik. Semakin banyak iterasi yang dilakukan, maka waktu yang dibutuhkan akan semakin lama.

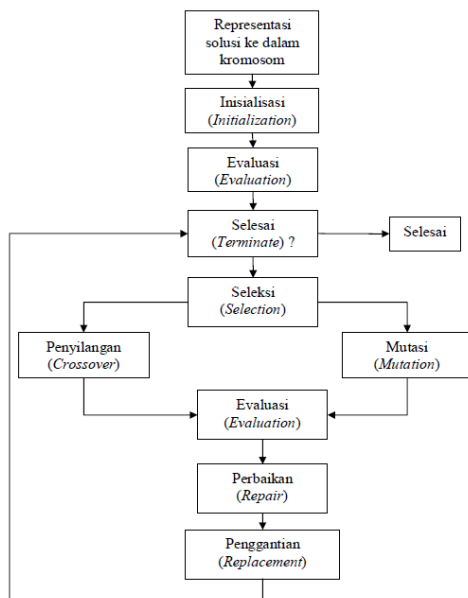
### 2.3 Prosedur Umum Algoritma Genetika

Prosedur umum algoritma genetika adalah sebagai berikut :

1. Pengkodean (*encoding*) calon solusi dan *set-up* beberapa parameter awal jumlah individu, probabilitas, penyilangan dan mutasi, dan jumlah generasi maksimum.

2. Pembangkitan acak sejumlah  $n$  kromosom pada generasi ke-0.
3. Evaluasi masing-masing kromosom dengan menghitung nilai *fitness*-nya.
4. Seleksi beberapa kromosom dari sejumlah  $n$  individu yang memiliki nilai *fitness* terbaik.
5. Rekombinasikan kromosom terpilih dengan cara melakukan penyilangan (*crossover*) dan mutasi (*mutation*).
6. *Update* jumlah generasi dan kembali ke langkah 2 sampai jumlah generasi maksimum tercapai.

Diagram alir algoritma genetika dapat dilihat pada gambar 1 :



**Gambar 1 : Diagram Alir Algoritma Genetika**

### Representasi atau Inisialisasi Kromosom

Inisialisasi kromosom direpresentasikan dalam bentuk larik dengan tipe data record yang berisi data yang mendukung proses penjadwalan. Panjang dari kromosom adalah sebanyak gen yang ada, dalam hal ini setiap gen mewakili mata kuliah yang ditawarkan. Setiap kromosom adalah barisan gen yang terdiri dari nilai hari, jam dan ruang.

### Fungsi *Fitness*

Fungsi *fitness* adalah fungsi yang akan mengukur tingkat kebugaran suatu

kromosom dalam populasi. Semakin besar nilai *fitness*, semakin bugar pula kromosom dalam populasi sehingga semakin besar kemungkinan kromosom tersebut dapat tetap bertahan pada generasi berikutnya.

Individu-individu dalam populasi telah terbentuk, maka langkah selanjutnya adalah menghitung nilai *fitness* setiap individu. Penghitungan dilakukan dengan memberikan pinalti untuk setiap aturan yang digunakan dalam penjadwalan. Semakin wajib aturan dilaksanakan, maka akan semakin besar nilai pinalti yang diberikan. Berikut aturan penghitungan fungsi *fitness* :

$$Fitness = \frac{1}{1 + Pinalti}$$

dimana :

$$Pinalti = \sum Bp \sum Np$$

Dari persamaan di atas nilai *fitness* ditentukan oleh nilai pinalti. Pinalti tersebut menunjukkan jumlah pelanggaran kendala pada suatu kromosom. Semakin tinggi nilai *fitness* akan semakin besar kemungkinan kromosom tersebut terpilih ke generasi berikutnya. Jadi nilai pinalti berbanding terbalik dengan nilai *fitness*, semakin kecil nilai pinalti (jumlah pelanggaran) semakin besar nilai *fitness*-nya.

Jadi fungsi *fitness* :

$$Fitness = \frac{1}{1 + \sum Bp \sum Np}$$

Keterangan :

$Bp$  : Bobot pelanggaran

$Np$  : Indikator pelanggaran

Aturan dan nilai pinalti dapat dilihat pada tabel berikut ini.

**Tabel 1 : Aturan dan Nilai Pinalti**

No.	Aturan	Nilai Pinalti
1	Sudah ada kelas yang sama, mata kuliah yang sama pada hari yang sama.	33

2	Terdapat kelas yang sama, jumlah sks yang sama pada mata kuliah yang sama tetapi memiliki dosen yang berbeda.	33
3	Sudah ada kelas lain untuk waktu yang sama, hari yang sama dan ruang yang sama.	11
4	Sudah ada kelas lain untuk waktu yang sama, hari yang sama dan kelas yang sama.	11
5	Sudah ada kelas lain untuk waktu yang sama, hari yang sama, dosen yang sama dan mata kuliah yang sama.	6
6	Sudah ada kelas lain untuk waktu yang sama, hari yang sama dan dosen yang sama.	5

### Seleksi (*Selection*) Kromosom

Setelah populasi awal terbentuk, setiap kromosom dalam populasi dievaluasi dengan menghitung nilai fungsi *fitness*-nya. Setelah itu proses pembentukan generasi baru diawali dengan seleksi kromosom. Seleksi adalah proses pemilihan beberapa kromosom untuk dijadikan sebagai kromosom induk bagi generasi berikutnya. Kromosom terpilih untuk masing-masing populasi di dalam generasi yang berikutnya berdasarkan nilai *fitness*.

### Penyilangan (*Crossover*) Kromosom

Setelah proses penyeleksian kromosom, langkah berikutnya adalah melakukan penyilangan terhadap pasangan-pasangan kromosom. Penyilangan (*crossover*) dikenal sebagai operator penggabungan ulang (*recombination*) yang paling utama dalam algoritma genetika. Penyilangan akan menukar informasi genetik antara dua kromosom induk yang terpilih dari proses seleksi untuk membentuk dua anak. Operator penyilangan bekerja pada sepasang kromosom induk untuk menghasilkan dua kromosom anak dengan menukarkan beberapa elemen (gen) yang dimiliki masing-masing kromosom induk.

Operator penyilangan biasanya dihubungkan dengan peluang penyilangan. Peluang penyilangan ( $P_c$ ) adalah rasio antara jumlah kromosom yang diharapkan

mengalami penyilangan dalam setiap generasi dengan jumlah kromosom total dalam populasi. Nilai  $P_c$  biasanya cukup tinggi (berkisar antara 0,6 – 1). Proses penyilangan akan terjadi pada sepasang kromosom jika suatu bilangan yang dibangkitkan secara acak ( $r$ ),  $0 \leq r \leq 1$ , nilainya kurang dari atau sama dengan  $P_c$ . Bilangan acak tersebut dibangkitkan setiap kali akan menyilangan sepasang kromosom. Tingkat penyilangan yang tinggi menyebabkan semakin besar kemungkinan algoritma genetika mengeksplorasi ruang pencarian sekaligus mempercepat ditemukannya solusi optimum. Peluang penyilangan yang tepat dan efektif hanya dapat diketahui melalui pengujian (*experiment*) khusus terhadap masalah yang bersangkutan. Misalnya ditentukan nilai  $P_c = 0.9$ .

### Mutasi (*Mutation*) Kromosom

Operasi ini akan menjadi sangat penting apabila nilai *fitness* kromosom dalam populasi cenderung sama atau sudah mencapai konvergen bias (*premature convergen*). Akibatnya, operator seleksi akan mengalami kesukaran memilih kromosom terbaik untuk dilakukan penyilangan. Dengan adanya operator mutasi, struktur suatu kromosom dapat dimodifikasi agar dihasilkan kromosom dengan struktur baru yang memiliki nilai *fitness* lebih baik.

Mutasi pada umumnya digunakan untuk mencegah tidak adanya kehilangan informasi sehingga dilaksanakan dengan pertukaran informasi di dalam kromosom. Operator mutasi dapat bersifat konstruktif maupun destruktif terhadap suatu kromosom, tetapi karena probabilitasnya yang kecil maka terjadinya mutasi akan sangat kecil karena didominasi oleh operator penyilangan. Peluang mutasi ( $P_m$ ) adalah rasio antara jumlah gen yang diharapkan mengalami mutasi pada setiap generasi dengan jumlah gen total dalam populasi. Nilai  $P_m$  yang digunakan biasanya sangat kecil (berkisar antara 0,001 – 0,2). Proses mutasi akan terjadi pada suatu gen, jika suatu bilangan yang dibangkitkan secara acak  $r$ ,  $0 \leq r \leq 1$ , nilainya kurang dari atau sama dengan  $P_m$ .

## 2.4 Metodologi Penelitian

Metode yang digunakan dalam pengembangan sistem aplikasi ini adalah *Rational Unified Process* (RUP). RUP memiliki empat buah tahap atau fase yang dapat dilakukan secara iteratif yaitu sebagai berikut :

### 1. Inception

Tahap ini dilakukan analisis sistem meliputi penentuan ruang lingkup serta analisis kebutuhan *user* melalui diagram *use case* dan analisis pemecahan masalah beserta sistem logika prosedur dengan diagram aktivitas.

### 2. Elaboration

Melalui hasil analisis pada tahap *inception*, maka pada tahapan berikutnya penulis mulai merancang sistem secara lengkap, mulai dari rancangan diagram kelas, deskripsi diagram *sequence*, perancangan perangkat lunak, tampilan antarmuka, dan algoritma program.

### 3. Construction

Pada tahap ini awalnya penulis menganalisis hasil dari desain aplikasi pada tahap *inception* dan *elaboration* dengan menentukan banyak tampilan dan kelas yang akan digunakan, setelah selesai maka penulis mulai mengimplementasikan pembuatan aplikasi pada perangkat keras.

### 4. Transition

Setelah perancangan aplikasi selesai diimplementasikan maka aplikasi diserahkan kepada *user* yang merupakan target dari pengembangan aplikasi dan melakukan pelatihan. Selanjutnya dilakukan uji coba terhadap aplikasi dengan meminta respon dari para *user* tentang aplikasi ini.

## 3 RANCANGAN PROGRAM

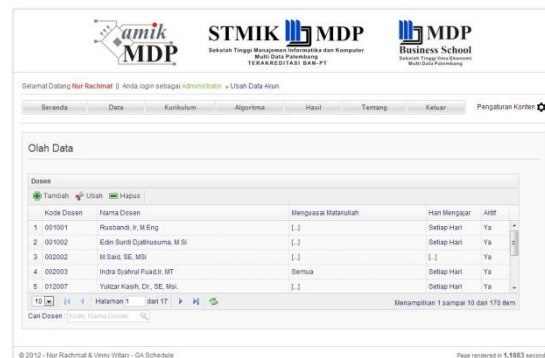
Rancangan data yang digunakan dalam aplikasi optimasi penjadwalan perkuliahan ini meliputi data dosen, mata kuliah, ruang, lantai, kelas, institusi, jurusan, tahun akademik, dan kurikulum.

Rancangan tampilan *login* dapat dilihat pada gambar 2.



Gambar 2 : Tampilan Login

Rancangan tampilan kelola data dapat dilihat pada gambar 3.



Gambar 3 : Tampilan Kelola Data

Rancangan tampilan proses algoritma dapat dilihat pada gambar 4.



Gambar 4 : Tampilan Proses Algoritma



- [9] Peranginangin, Kasiman 2006, *Aplikasi Web dengan PHP dan MySQL*, Andi Offset, Yogyakarta.
- [10] Rational 2001, *Rational Unified Process Best Practices for Software Development Teams*. Diakses 11 Oktober 2012, dari <http://www.ibm.com>
- [11] Sidik, Betha 2012, *Framework CodeIgniter*, Informatika Bandung, Bandung.
- [12] Sutanta, Edhy 2011, *Basis Data dalam Tinjauan Konseptual*, Andi, Yogyakarta.
- [13] Sutojo, T., Edi Mulyanto dan Vincent Suhartono 2011, *Kecerdasan Buatan*, Andi Offset, Yogyakarta.
- [14] Syarif, A. dan Mitsuo G. 2003, *Hybridized Parallel Genetic Algorithm for Facility Location Problem*, Makalah Penelitian IlmuKomputer.com.
- [15] W. Pandjaitan, Lanny 2007, *Dasar-dasar Komputasi Cerdas*, Andi, Yogyakarta.